What Is Claimed Is:

1. A method for reconstructing a sequence of processes of a control program executable by one of a computing device from contents of a first table and a second table, the control program being subdivided into a plurality of tasks, each of the plurality of tasks including at least one process, the method comprising:

storing, during execution of the control program and for each completed task, an identifier for one of the at least one process last executed before starting the completed task, in the first table;

storing an order of respective ones of the completed tasks in the second table;

generating at first, from the contents of the first table and the second table, a third table containing, for a new task, the identifier for one of the at least one process last executed before the start of the new task; and

reconstructing a complete process sequence of the control program from the third table using the order of execution of the at least one process of the plurality of tasks.

2. The method of claim 1, wherein:

to create the third table, the identifiers for respective last processes of the plurality of tasks stored in the second table are first stored in the third table;

for each identifier in the first table, it is checked whether a corresponding process is the last process of a corresponding one of the plurality of tasks, and

if the identifier corresponds to the last process of the corresponding one of the plurality of tasks, no entry is made in the third table, and

if the identifier does not correspond to the last process of the corresponding one of the plurality of tasks, a checked identifier is stored in the third table before the identifier for the first process, contained in the first table, of one of the plurality of tasks that

was completed at a position corresponding to another position of the checked identifier in the first table.

3. The method of claim 2, wherein to determine the identifier for the first process of the plurality of the tasks, contained in the first table, a fourth table, for each of the plurality of tasks, is operable to store information as to whether a particular task has already begun, and contents of the fourth table are checked for a completed task that was completed at a position corresponding to the another position of the checked identifier in the first table.

4. The method of claim 3, wherein a memory cell is set in the fourth table for one of the plurality of tasks as soon as a particular process of the plurality of tasks that is the first to be interrupted by another task is encountered during reconstruction of the process sequence, and the memory cell is cleared as soon as the last process of the plurality of tasks is encountered during reconstruction of the process sequence.

5. The method of claim 2, wherein, determining the first process, interrupted by another one of the plurality of tasks, of the plurality of tasks that was completed at the position corresponding to the position of the checked identifier in the first table, includes:

using a fifth table to store information as to whether processes stored in the third table are the first processes of corresponding ones of the plurality of tasks interrupted by another one of the plurality of tasks, and

checking contents of the fifth, for processes preceding the position of the checked identifier in the third table, to find whether these processes are the first processes, interrupted by another one of the plurality of tasks, of a completed task that was completed at a position corresponding to the position of the checked identifier in the first table.

6. The method of claim 5, wherein a memory cell is set in the

fifth table as soon as, during reconstruction of the process sequence, a particular process stored in the third table is encountered that is the first process of a corresponding one of the plurality of tasks to be interrupted by another one of the plurality of tasks.

7. The method of claim 1, wherein the reconstructing of the complete process sequence includes:

checking the identifiers in the third table in a direction opposing the process sequence to determine one of whether the process corresponding to a checked identifier belongs to one of the plurality of tasks having only one process, and whether the process corresponding to the checked identifier is the first process of the corresponding task contained in the first table; and

storing the identifiers for checked processes in a direction opposite a process sequence in a one-dimensional seventh table, and

one of:

if one of a process corresponding to the checked identifier belongs to a task having only one process, and the process corresponding to the checked identifier is the first process of the corresponding one of the plurality of tasks to be interrupted by another one of the plurality of tasks, no entry is made in the seventh table, and

if a process corresponding to the checked identifier belongs to a task having a plurality of processes, and the process corresponding to the checked identifier is not the process of the corresponding one of the plurality of tasks that is the first to be interrupted by another one of the plurality of tasks, the third table searched for the identifier for the process of the corresponding task that precedes the checked identifier, starting from the position of the checked identifier in a direction opposite the process sequence, and

if at least one process is missing before the process

corresponding to the checked identifier, the identifier for the at least one missing process is entered in the seventh table before the process corresponding to the checked identifier.

8. A memory element storing a computer program executable on a computing device to perform a method for reconstructing a sequence of processes of a control program executable by one of a computing device from contents of a first table and a second table, the control program being subdivided into a plurality of tasks, each of the plurality of tasks including at least one process, the method including:

storing, during execution of the control program and for each completed task, an identifier for one of the at least one process last executed before starting the completed task, in the first table;

storing an order of respective ones of the completed tasks in the second table;

generating at first, from the contents of the first table and the second table, a third table containing, for a new task, the identifier for one of the at least one process last executed before the start of the new task; and

reconstructing a complete process sequence of the control program from the third table using the order of execution of the at least one process of the plurality of tasks.

9. A computer program executable on a computing device to perform a method for reconstructing a sequence of processes of a control program executable by one of a computing device from contents of a first table and a second table, the control program being subdivided into a plurality of tasks, each of the plurality of tasks including at least one process, the method including:

storing, during execution of the control program and for each completed task, an identifier for one of the at least one process last executed before starting the completed task, in the first table;

storing an order of respective ones of the completed tasks in the second table;

generating at first, from the contents of the first table and the second table, a third table containing, for a new task, the identifier for one of the at least one process last executed before the start of the new task; and

reconstructing a complete process sequence of the control program from the third table using the order of execution of the at least one process of the plurality of tasks.

10. The computer program of claim 9, wherein the computer program is stored in a memory element.

11. A device for reconstructing a sequence of processes of a control program executable by one of a computing device from contents of a first table and a second table, the control program being subdivided into a plurality of tasks, each of the plurality of tasks including at least one process, the device comprising:

a first storing arrangement to store, during execution of the control program and for each completed task, an identifier for one of the at least one process last executed before starting the completed task, in the first table;

a second storing arrangement to store an order of respective ones of the completed tasks in the second table;

a generating arrangement to generate at first, from the contents of the first table and the second table, a third table containing, for a new task, the identifier for one of the at least one process last executed before the start of the new task; and

a reconstructing arrangement to reconstruct a complete process sequence of the control program from the third table using the order of execution of the at least one process of the plurality of tasks.

12. The method of claim 11, wherein the computing device includes a microprocessor.

13. The device of claim 1, wherein the computing device includes a microprocessor.

14. The memory element of claim 8, wherein the computing device includes a microprocessor.

15. The memory element of claim 8, wherein the memory element includes a flash memory.

16. The computer program of claim 9, wherein the computing device includes a microprocessor.

17. The computer program of claim 9, wherein the memory element includes a flash memory.